Centre de recherche sur les transports (C.R.T.)
Centre for Research on Transportation

CRT-2003-27

MAXIMIZING THE VALUE OF AN EARTH OBSERVATION SATELLITE ORBIT

by

Jean-François Cordeau

Gilbert Laporte

October 2003

Université de Montréal
C.P. 6128, succursale Centre-ville, Montréal QC H3C 3J7 Canada
Téléphone: (514) 343-7575 / Télécopieur: (514) 343-7121

crt@crt.umontreal.ca
www.crt.umontreal.ca

HEC ÉCOLE POLYTECHNIQUE MONTRÉAL Université de Montréal

104-00343/

PUBLICATION
CRT-2003-27

MAXIMIZING THE VALUE OF AN EARTH OBSERVATION SATELLITE ORBIT

by

Jean-François Cordeau[1]
Gilbert Laporte[1]

This research was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

[1] Centre de recherche sur les transports, Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal, Canada H3C 3J7 and Canada Research Chair in Distribution Management, HEC Montréal, 3000 Chemin de la Côte-Sainte-Catherine, Montréal, Canada H3T 2A7

Abstract

Earth observation satellites are platforms equipped with optical instruments that orbit the planet. During the course of an orbit they take photographs of some regions of the Earth at the request of customers. Each photograph generates a profit but, due to presence of several constraints, not all requests can be satisfied. The problem is to select a subset of requests of maximal profit for a given orbit. The problem is solved by means of a tabu search heuristic and computational results are reported. This work was initiated as part of a challenge organized by the French Operational Research Society. The algorithm won the second prize in the final round of the competition.

Key words: satellites, selective traveling salesman problem, knapsack problem, tabu search heuristic.

Résumé

Les satellites d'observation de la terre sont des plateformes équipées d'instruments optiques qui réalisent des orbites autour de la planète. Au cours de chaque orbite, ils prennent des photos de différentes régions de la terre à la demande des clients. Chaque photo génère un profit mais la présence de plusieurs contraintes fait en sorte que toutes les requêtes ne peuvent être satisfaites. Le problème consiste à choisir un sous-ensemble de requêtes de profit maximal pour une orbite donnée. Le problème est résolu par une heuristique de recherche avec tabous et des expériences de calcul sont présentées. Ce travail a été réalisé dans le cadre d'une compétition organisée par l'Association française de recherche opérationnelle et d'aide à la décision. L'algorithme a été classé deuxième dans la ronde finale de la compétition.

Mots-clés : satellites, problème du voyageur de commerce sélectif, problème de sac alpin, heuristique de recherche avec tabous.

Introduction

Earth observation satellites are platforms equipped with optical instruments that orbit the planet. Over a period of several days they perform a cycle of orbits of the Earth. Each orbit is phased out with respect to the preceding one but the trajectory is cyclic in the sense that the satellite recovers its initial position after a number of orbits. A full cycle enables the satellite to view each area of the planet. In the course of their mission satellites take photographs of specific areas of the Earth at the requests of a number of users including governments, research institutes, and companies. Each request generates a profit. Typically, the number of requests exceeds what can feasibly be accommodated during a mission. The problem is to select a subset of requests yielding a maximal profit, subject to operational constraints. The present study focuses on maximizing the profit associated with a single orbit of the satellite. This problem is called the *Satellite Orbit Problem* (SOP).

A request can either be a target or a polygon. A target consists of a circle of limited dimension whereas a polygon may cover a wide geographical area. Because of their size, polygons cannot usually be photographed in a single shot. For this reason, they are partitioned into strips of equal width but possibly unequal lengths (Figure 1). The time required to photograph or *acquire* a strip is proportional to its length. In the course of a single orbit, the satellite may be able to photograph several strips of the same polygon by rotating the camera between consecutive shots. For our purpose a target can be seen as a polygon comprising a single strip. As the satellite approaches a strip it has a *forward view* of it. After it has passed the strip, it has a *backward view* of it. Some requests are *mono* while others are *stereo*. A mono request consists of a single shot of each strip in the polygon. A stereo request consists of two shots of each strip at different angles, but in the same direction, that is two forward views or two backward views. A strip from a stereo request is considered to have been acquired only if its twin strip has also been acquired. In the sequel the acquisition of a target or of a strip in a given direction will be called a *job*. Two jobs associated with the same strip from a stereo request will be called *twin* jobs.

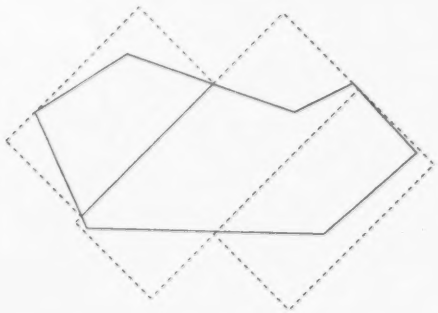

Figure 1: A polygon partitioned into three strips

A total of n jobs are considered. Each job i has a duration d_i and must be executed within a time window $[a_i, b_i]$ which corresponds to the interval during which a target or a strip is

within the field of view of the satellite. The transition time c_{ij} from job i to job j depends on the time required to move the camera at the appropriate angle to perform job j after job i has been completed. Because time windows are moderately large, there often exist several feasible orderings with different total transition times for a given subset of jobs. The problem thus consists not only of selecting which jobs to perform but also of determining their sequence and the time at which each of them should be executed.

The objective function is the sum of the gains associated with the acquisition of the targets and polygons. The profit associated with a polygon is a piecewise linear convex function of the surface of that polygon that is acquired (Figure 2). This reflects the fact that partial images have little value to the user.

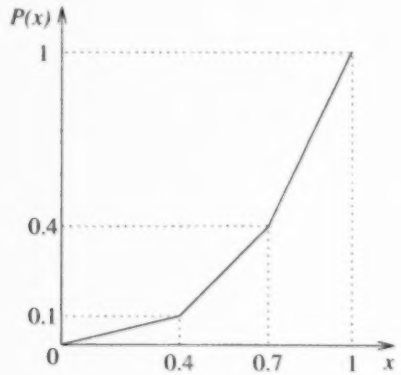

Figure 2: Non-linear profit function $P(x)$ associated with the acquired portion x of a polygon

Our aim is to present a tabu search heuristic for the SOP. This work was initiated within a challenge organized by the French Operational Research Society (ROADEF) for its February 2003 meeting. In total 18 proposals were submitted for the qualification stage and our algorithm won the second prize in the final stage. The background information provided in this introduction was extracted from two documents[1,2] distributed to the challengers. The remainder of this article is organized as follows. A short literature review is presented in the next section. This is followed by a description of our tabu search algorithm, by computational results, and by the conclusion.

Literature review

Relative to other combinatorial optimization problems the SOP has only received limited attention. Also, a number of variants of the problem have been introduced, making methodological comparisons difficult. A related problem, described by Hall and Magazine[3], is the maximization of the value of a space mission. Here one must determine the scientific agenda of a spacecraft by selecting projects from among a list. As in the SOP each project has an

associated profit and can only be carried out within a preset time window. The problem is to select a subset of projects whose total profit is maximal, respecting the time window constraints and the duration of the space mission. The authors propose an integer linear programming formulation for this problem. They also describe eight heuristic procedures as well as two upper bounding procedures. These procedures are combined into a dynamic programming algorithm that can solve to optimality randomly generated instances with up to 200 requests. A similar problem, arising this time in the area of an Earth observation satellite mission, is described by Wolfe and Sorensen[4]. The main difference with the previous study lies in the definition of the objective function. The profit associated with a request is defined by means of a so-called suitability function which takes a higher value if the request is executed close to the middle of its time window. The authors describe two simple construction heuristics as well as a genetic algorithm for their problem. They present computational results on randomly generated instances involving up to 50 requests.

The three publications by Bensana et al.[5] and Vasquez and Hao[6,7] treat a problem closer to ours. The first of these articles describes the problem and proposes a set of benchmark instances and lower bounds for the daily management of *SPOT* satellites. These satellites make 14 Earth orbits per day and complete a cycle in 26 days. As a result, the number of requests to be considered for a given day may become very large. Vasquez and Hao[6] study uncapacitated and capacitated versions of this problem. The capacity constraint relates to the total information that can be recorded in the satellite in a given day. The authors view the problem as a knapsack problem with side constraints and develop a tabu search algorithm that integrates constraint relaxation and dynamic tabu tenure mechanisms. Experiments were carried out on uncapacitated instances with $67 \leq n \leq 364$ and capacitated instances with $209 \leq n \leq 1057$. In the uncapacitated case, the known optimal value was attained for all instances. In the capacitated case, no optimal values were known except for one instance. Comparisons were therefore made with best known solution values and improvements could be obtained for all instances. Finally, the second Vasquez and Hao[7] article describes upper bounds for single-orbit and multiple-orbit instances. The upper bound values are relatively tight (between 0% and 3.45% of the optimum on a set of benchmark instances) and allow the authors to conclude that their tabu search heuristic produces high quality results.

Algorithm

The SOP bears several similarities with the *Selective Traveling Salesman Problem* (STSP) encountered in the vehicle routing literature[8]. The STSP is itself a variant of the *Traveling Salesman Problem* in which each vertex i carries a profit p_i and the objective is to construct a tour of bounded length c over a subset of vertices in order to maximize profit. The SOP belongs to the same family as the STSP but has different features, namely 1) its objective function is non-linear; 2) it contains linking constraints on the requests; and 3) time windows are imposed.

The heuristic we have developed for the SOP borrows some elements from the *Unified Tabu Search Algorithm* (UTSA) developed by Cordeau, Laporte and Mercier[9] for the *Vehicle*

Routing Problem with Time Windows (VRPTW). Our experience has shown that UTSA is not only a powerful tool for the VRPTW, but is also higly flexible and can be adapted to a host of other routing problems. We now provide a brief description of its adaptation to the SOP.

Our algorithm is based on tabu search. It explores the solution space by moving at each iteration from the current solution s, defined as a sequence of jobs, to the best solution in its neighbourhood $N(s)$. Since the solution may deteriorate between two successive iterations, an anti-cycling mechanism is put in place by declaring tabu attributes of recently visited solutions.

Constraint relaxation

An important feature of our algorithm is the possibility of exploring infeasible solutions during the search. In the SOP we only allow violations of time window constraints. Thus the value of solution s is defined as $f(s) = p(s) - \alpha w(s)$, where $p(s)$ is the profit associated with s and $w(s)$ is the total time window violations. The parameter α is initially set equal to 1 and self-adjusts during the course of the search to allow a mix of feasible and infeasible solutions. More precisely, at each iteration the value of α is multiplied by $1 + \delta$, with $\delta \geq 0$, if the current solution is infeasible and divided by $1 + \delta$ otherwise. In our implementation, the value of δ is selected randomly in the interval $[0, 1]$ at each iteration.

Neighbourhood structure

At each iteration, the following six types of moves are considered for all jobs to define the neighbourhood $N(s)$ of the current solution s:

1. Insert a mono job i in the solution.
2. Remove a mono job i from the solution.
3. Insert twin jobs i and j in the solution.
4. Remove twin jobs i and j from the solution.
5. Replace mono job i with mono job j where i and j correspond to the same strip photographed in opposite directions.
6. Replace twin jobs i and j with twin jobs k and l where i and j are twin jobs, and k and l correspond to the same strips as i and j photographed in the opposite direction.

Starting from the trivial solution in which no job is selected, these moves maintain at each iteration the feasibility of the solution with respect to all constraints but the time windows.

When inserting a mono job in the solution, the insertion position is chosen so as to maximize the value of $f(s)$ by taking into account the increase in time window violations. This is accomplished by performing a simple insertion. In other words, the ordering of the jobs that are already in the solution remains the same. Even though solutions violating time windows

are allowed during the search, computation times are reduced by considering the insertion of a job between two successive jobs only if it is possible to satisfy the time windows of all three jobs with this sequence. Of course, the time windows of jobs that come after these three may become violated after performing the insertion.

When handling twin jobs, the first job is inserted in its best possible position and, keeping this position fixed, the second job is then inserted in its best position. Treating each of the two jobs as the first yields two different ways of performing these operations and both are evaluated to identify the best insertion. Replacing mono or twin jobs (in exchanges of type 5 or 6) is done similarly by first removing the jobs to be replaced and then inserting the new ones.

To evaluate the impact of a move, the time variables associated with all jobs in the solution must be recomputed in order to measure the total impact on time window violations.

Tabu tenures

The tabu mechanism is implemented as follows. Whenever a job is removed from the solution, its reinsertion is forbidden for θ iterations, where the value of θ is drawn randomly from the interval $[0, \sqrt{n/2}]$ and rounded to the nearest integer. Through an aspiration criterion, the tabu status of a job can, however, be revoked if that would allow the search process to reach a solution of larger profit than that of the best solution found in which this job was present.

Diversification

The algorithm uses two diversification mechanisms. The first is a continuous diversification scheme now common to several tabu search implementations. Any solution $\bar{s} \in N(s)$ such that $f(\bar{s}) \leq f(s)$ is penalized by a factor proportional to the frequency of its distinguishing job and a scaling factor. More precisely, let ρ_k be the number of iterations during which job k has been part of the solution since the start of the search process and let η be the total number of iterations performed. If moving from solution s to solution $\bar{s}$ requires the insertion of job k in the sequence, then a penalty $q(\bar{s}) = \lambda\, p(\bar{s})\, \rho_k/\eta$ is subtracted from $f(\bar{s})$. The scaling factor $p(\bar{s})$ introduces a correction to adjust the penalties with respect to total solution cost. The parameter λ, equal to $\log_{10}(n/2)$ in our implementation, controls the intensity of the diversification. These penalties drive the search process toward less explored regions of the search space whenever a local optimum is reached.

The second diversification mechanism perturbs the solution under certain circumstances. If the best known solution has not improved for 100μ iterations, where μ is the number of perturbations made since the last improvement, then the search stops and restarts from the best known solution s^*. However, solution s^* is perturbed by removing a proportion π of randomly selected jobs where π is itself chosen randomly in the interval $[0, 1]$. When a job from a stereo request is selected for removal, its twin job is also removed. The reinsertion of these jobs is then declared tabu for θ iterations. The value of μ is initially set equal to 1 and reset to 1 whenever a new best solution has been identified.

Intra-sequence exchanges

Every 100 iterations, a reordering of the jobs in the solution is performed in the hope of improving feasibility. Each job currently in the solution is removed from the sequence and reinserted in its best position in order to minimize $w(s)$. Again, these exchanges maintain feasibility of all constraints except time windows.

Restarts

Finally, the algorithm uses false starts to avoid being trapped in a poor local optimum because of the moves performed in the first few iterations. Specifically, the tabu search procedure is run five times for $100 \ln(n/2)$ iterations and the best solution identified during these runs is used as the starting solution for the main search.

Implementation and computational results

The tabu search algorithm just described was implemented in C and tested on a set of 20 instances provided by the challenge organizers. The instances are divided into two subsets of 10 instances. Table 1 indicates, for each instance, the total number of requests (Reqs) which are divided into mono targets (MT), stereo targets (ST), mono polygons (MP) and stereo polygons (SP). The TS column indicates the number of targets and srips in these requests. The number of jobs, n, is twice as large because of the two possible directions of acquisition.

Table 1: Characteristics of the test instances

	Subset A							Subset X					
Instance	Reqs	MT	ST	MP	SP	TS	Instance	Reqs	MT	ST	MP	SP	TS
2_13_111	68	52	12	4	0	106	2_28_111	270	201	51	15	3	408
2_15_170	218	167	39	12	0	295	2_28_140	275	211	48	13	3	379
2_26_96	336	255	53	26	2	483	2_28_155	321	231	68	19	3	470
2_27_22	375	287	59	25	4	534	2_28_170	302	223	56	19	4	468
2_9_170	12	9	1	1	1	25	2_28_37	327	232	69	23	3	522
2_9_36	2	2	0	0	0	2	2_28_66	242	168	49	22	3	375
2_9_66	4	3	0	1	0	7	2_28_7	305	230	52	21	2	428
3_25_22	150	16	1	47	86	342	2_28_81	317	241	57	16	3	446
3_8_155	12	0	0	3	9	28	3_28_155	128	8	0	34	86	305
4_17_186	77	23	0	14	40	147	3_28_96	128	9	0	39	80	292

Subset A was used by the participants to develop and fine-tune their algorithm in a qualification stage. A total of 18 teams submitted an algorithm and 11 of them qualified for the final stage. The selected teams were then allowed to submit an improved version of their algorithm which was ranked by the organizers based on the instances from a new subset X. The score of a team was computed as the average percentage deviation between the solutions produced by the proposed algorithm and the reference solutions provided by the organizers. The winning algorithm was that of Kuipers[10] with a score of 31.76 while our algorithm came close second with a score of 30.84.

Tables 2 and 3 report the reference solution values provided by the organizers as well as the average and best solution values obtained by the latest versions of the algorithm of Kuipers and ours (the values in parentheses indicate the percentage deviation from the reference solutions). These average and best values were obtained by running each algorithm 10 times on a Sun Blade 1000 computer and allowing it to run for 300 seconds each time. The last line of each table reports the scores computed with the formula used by the organizers.

These results show that our algorithm was able to identify very good quality solutions on each of the twenty instances. In particular, the best solutions found by our algorithm are very close to those of Kuipers. Finally, in each case where our algorithm did not find the best known solution within the time limits set by the organizers, additional experiments performed afterwards with slightly larger computation times have allowed us to obtain solutions of equal profit to those produced by Kuipers.

Conclusion

We have described and solved a hard combinatorial optimization problem arising in the planing of Earth observation satellite missions. The problem was solved by means of a tabu search heuristic. This heuristic is rather simple, both conceptually and from an implementation point of view. It has enabled its authors to win the second prize in the ROADEF competition. Work is currently under way in collaboration with the French CNES (Centre national d'études spatiales) to adapt the heuristic to more complicated situations involving multiple orbits.

Acknowledgments

We thank the members of the ROADEF challenge organizing committee, Van-Dat Cung, Gérard Verfaillie, and Nicolas Bataille. This work was supported by the Canadian Natural Science and Engineering Research Council under grants 227837-00 and OGP0039682.

Table 2: Results on instances from subset A (qualification stage)

		Kuipers		**Cordeau and Laporte**	
Instance	Reference	Average	Best	Average	Best
2_13_111	539347530	563597071 (4.50)	563597071 (4.50)	552363210 (2.41)	563597071 (4.50)
2_15_170	524896372	716817920 (36.56)	717383629 (36.67)	710599718 (35.38)	717239040 (36.64)
2_26_96	671624710	1004434419 (49.55)	1005301900 (49.68)	975605140 (45.26)	1005301900 (49.68)
2_27_22	615164930	965414720 (56.94)	967910750 (57.34)	951323943 (54.65)	966056500 (57.04)
2_9_170	191358231	191358231 (0.00)	191358231 (0.00)	191358231 (0.00)	191358231 (0.00)
2_9_36	10423440	10423440 (0.00)	10423440 (0.00)	10423440 (0.00)	10423440 (0.00)
2_9_66	115710959	115710959 (0.00)	115710959 (0.00)	115710959 (0.00)	115710959 (0.00)
3_25_22	333087930	425983220 (27.89)	425983220 (27.89)	425982756 (27.89)	425983220 (27.89)
3_8_155	121680360	121680360 (0.00)	121680360 (0.00)	121680360 (0.00)	121680360 (0.00)
4_17_186	175218520	185406780 (5.81)	185406780 (5.81)	185406780 (5.81)	185406780 (5.81)
Avg.		(18.13)	(18.19)	(17.14)	(18.16)

Table 3: Results on instances from subset X (final stage)

		Kuipers		Cordeau and Laporte	
Instance	Reference	Average	Best	Average	Best
2_28_111	641972335	872246792 (35.87)	875447480 (36.37)	863378847 (34.49)	870235760 (35.56)
2_28_140	688988965	833061980 (20.91)	833286609 (20.94)	830212363 (20.50)	833286609 (20.94)
2_28_155	754865230	952267030 (26.15)	952267030 (26.15)	941946991 (24.78)	952267030 (26.15)
2_28_170	643689577	952144183 (47.92)	957935459 (48.82)	945346868 (46.86)	963809499 (49.73)
2_28_37	632907795	987549761 (56.03)	992155179 (56.76)	962143842 (52.02)	975346580 (54.11)
2_28_66	686598355	945409350 (37.69)	945737319 (37.74)	943683278 (37.44)	945737319 (37.74)
2_28_7	714903677	977811340 (36.78)	977811340 (36.78)	971491820 (35.89)	977811340 (36.78)
2_28_81	728168195	878847950 (20.69)	878847950 (20.69)	872464834 (19.82)	878847950 (20.69)
3_28_155	380503800	457063785 (20.12)	462070340 (21.44)	462070340 (21.44)	462070340 (21.44)
3_28_96	396704166	458107362 (15.48)	458107362 (15.48)	456688588 (15.12)	458107362 (15.48)
Avg.		(31.76)	(32.12)	(30.84)	(31.86)

References

[1] G. Verfaillie, M. Lemaître, N. Bataille, and J.-M. Lachiver. Gestion de la mission de satellites d'observation de la terre. Description informelle du problème global. Technical report, 2002.

[2] G. Verfaillie, M. Lemaître, N. Bataille, and J.-M. Lachiver. Gestion de la mission de satellites d'observation de la terre. Description du challenge. Technical report, 2002.

[3] N.G. Hall and M.J. Magazine. Maximizing the value of a space mission. *European Journal of Operational Research*, 78:224-241, 1994.

[4] W.J. Wolfe and S.E. Sorensen. Three scheduling algorithms applied to the Earth observing systems domain. *Management Science*, 46:148-168, 2000.

[5] E. Bensana, M. Lemaître, and G. Verfaillie. Earth observation satellite management. *Constraints*, 4:293-299, 1999.

[6] M. Vasquez and J.-K. Hao. A "Logic-constrained" knapsack formulation and a tabu algorithm for the daily photograph scheduling of an Earth observation satellite. *Computational Optimization and Applications*, 20:137-157, 2001.

[7] M. Vasquez and J.-K. Hao. Upper bounds for the SPOT 5 daily photograph scheduling problem. *Journal of Combinatorial Optimization*, 7:87-103, 2003.

[8] M. Gendreau, G. Laporte, and F. Semet. A branch-and-cut algorithm for the undirected selective traveling salesman problem. *Networks*, 32:263–273, 1998.

[9] J.-F. Cordeau, G. Laporte, and A. Mercier. A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Society*, 52:928–936, 2001.

[10] E.J. Kuipers. Algorithm for the management of the missions of Earth observation satellites. Booklet of abstracts, Challenge ROADEF 2003.